A Tool for Semi-Automatic and Interactive Annotation of Dialogue Utterances with Information States

David Schlangen and **Alex Lascarides** and **Jason Baldridge**
School of Informatics
University of Edinburgh
{das|alex|jmb}@cogsci.ed.ac.uk

1 Introduction

The availability of linguistically annotated corpora like the Penn Treebank has long proven beneficial for computational linguistics research. However, attempts have begun only recently to provide corpora with *discourse* information (c.f. e.g. URML (Reitter and Stede, 2003) or the Penn Discourse Treebank (PDTB) project at the University of Pennsylvania[1]). Here we report on a project whose goal is to generate a corpus annotated with deep discourse semantic information which can then be used to train statistical models of semantic interpretation. We introduce our general methodology and describe how it is instantiated in a purpose-built tool that supports interactive, semi-automated annotation. The novel feature of this tool is its use of a reasoning engine which implements a semantic theory of discourse interpretation to suggest annotations to the user.[2]

2 Overview of the System

At the core of our annotation system lies a module which provides an infrastructure for interactively annotating dialogues. It provides routines to read (XML-formatted) dialogues, pass them to a processing engine, let users edit and possibly reprocess the results of that step, and finally store the finished annotation (also XML-formatted). This infrastructural module is called ANT, for *Annotation Tool*.

This part of the system is independent from any particular annotation task; the modules that make use of the infrastructure on the other hand have to be tailored to the task at hand. For example, for any given task the exact input and output format has to be specified, and a discourse-processing module which computes suggestions for annotations has to be provided. Figure 1 shows an instantiation of the architecture where the discourse processing is done by a system called RUDI (*Resolving Underspecification using Discourse Information*, cf. (Schlangen and Lascarides, 2002)) and the input consists of semantic representations computed by a wide-coverage HPSG (the *English Resource Grammar* (ERG); disambiguated parses of the input utterances are provided by the REDWOODS treebank (Oepen et al., 2002)). This instantiation of the system is called RANT (for RUDI-*based* ANT).

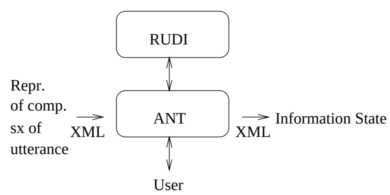

Figure 1: Schematic Overview of RANT

In general terms, an annotation cycle in ANT proceeds as follows:
(a) First, one basic unit of the data that is to be annotated is read in; this could be an orthographic transcription of an utterance in a dia-

[1] C.f. http://www.cis.upenn.edu/~dltag/

[2] Thus the development of the tool also allows us to evaluate and improve the symbolic theory underlying the computation of suggestions by exposing it to large sets of "real-world" data.

logue, together with a representation of its syntactic and semantic structure. (b) This information is passed to the discourse processing module, together with previously collected information (the *context*). The system expects back from that module an *information state* (IS) representation, which might for example contain information about performed speech acts, or about changes in the common ground of the dialogue participants. In general, any information state theory of dialogue could be utilised here. (c) The IS is presented in a GUI to the user/annotator, who can then edit it. This process is interactive, because at any point the revised information state can be sent back to the processing module in order to compute consequences of the changes. (d) Once the annotator is satisfied with the IS, it is stored, and a new annotation cycle begins with the current IS as part of the *context*.

3 Extending the REDWOODS Treebank

We use RANT, an instantiation of this architecture, to create a treebank annotated with discourse information. The basis of this is the REDWOODS treebank (Oepen et al., 2002), which provides disambiguated parses for approximately 7000 dialogue-utterances from the domain of appointment scheduling. The kind of discourse information we are interested in and for which suggestions are computed by RUDI can be illustrated with the following example. (The h_i in parentheses are the *labels* of the utterances.)

(1) (h_1) A: What is a good time for you?
(h_2) B: After 2pm on Monday…
(h_3) …and I'm also free on Wednesday.

The IS with which we annotate the utterances consists of two main elements: a logical form for the dialogue in terms of a *discourse structure*, and (parts of) the *model* that satisfies that structure. This discourse structure consists of the following elements: (i) the grouping of the utterances into larger discourse units (e.g., utterances h_2 and h_3 in (1) are grouped together); (ii) rhetorical relations connecting these segments (e.g., h_3 is a *continuation* of h_2, and the resulting segment provides an indirect answer to question h_1); and (iii) resolutions of (some) underspecification in the logical forms, namely that arising from the use of fragments (e.g., h_2 is resolved to something paraphraseable by "After 2pm on Monday is a good time…"), and that arising from the need to bridge definites to the context (e.g., "Wednesday afternoon" in h_3 is resolved to be the next Wednesday afternoon after the time of utterance). These three elements of discourse structure are logically dependent; for example, a particular rhetorical relation can have truth conditional consequences which constrain the values of bridging relations among temporal referents.

The IS also records certain parts of models of the discourse structure, namely the denotations of temporal referents—defined as intervals on the calendar—that make the discourse structure true, given knowledge about when the conversation took place. Finally, it also records the purpose behind each utterance, insofar as the overall goal of finding a time to meet is concerned.

4 Related Work

While we use the system with a particular dialogue theory as background, we expect ANT to be useful for evaluating and developing all kinds of IS-based theories. In contrast to URML and PDTB mentioned above we aim to annotate our corpus with much more detailed semantic information such as goals and resolutions of semantically underspecified information described in Section 3, expecting that this additional information will prove useful for training statistical models.

References

S. Oepen, D. Flickinger, K. Toutanova, and C.D. Manning. 2002. LinGO redwoods. a rich and dynamic treebank for HPSG. In *Proceedings of the 1st Workshop on Treebanks and Linguistic Theories*, pages 139–149, Sozopol, Bulgaria.

D. Reitter and M. Stede. 2003. Step by step: Underspecified markup in incremental rhetorical analysis. In *Proceedings of the 4th International Workshop on Linguistically Interpreted Corpora*, Budapest.

D. Schlangen and A. Lascarides. 2002. Resolving fragments using discourse information. In J. Bos, M.E. Foster, and C. Matheson, editors, *Proceedings of EDILOG 2002*, pages 161–168, Edinburgh, September.